	Type	L #	Hits	Search Text	DBs
1	BRS	L1	170	dictionary with (compress$3 or encod$3 or cod$3) same (color or colour)	USPAT
2	BRS	L2	6	dictionary with (compress$3 or encod$3 or cod$3) same ((color or colour) with pixel)	USPAT
3	BRS	L3	0	dictionary with (compress$3 or encod$3 or cod$3) same ((color or colour) with pixel)	EPO
4	BRS	L4	1	dictionary with (compress$3 or encod$3 or cod$3) same ((color or colour) with pixel)	JPO
5	BRS	L5	1	dictionary with (compress$3 or encod$3 or cod$3) same ((color or colour) with pixel)	DERWENT

	Type	L #	Hits	Search Text	DBs
1	BRS	L1	0	palettised with image with compression	USPAT
2	BRS	L2	33	palet$8 with image with compression	USPAT
3	BRS	L3	2	2 and (dictionar$3 or lzw)	USPAT
4	BRS	L4	0	(palet$8 with image with compression) same (lzw or dictionary)	USPAT
5	BRS	L5	0	(palet$8 with image with compression) same (lzw or dictionar$6)	USPAT
6	BRS	L6	19	(palet$8 with image with compression) same ((color colour) with pixel)	USPAT
7	BRS	L7	1	6 and (dictionar$5 or lzw)	USPAT
8	BRS	L8	1	(palet$8 with image with compression) same ((color colour) with pixel)	EPO
9	BRS	L9	0	(palet$8 with image with compression) same ((color colour) with pixel)	DERWENT
10	BRS	L10	0	(palet$8 with image with compression) same ((color colour) with pixel)	JPO

Your search matched **5** of **1105713** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

(compres* or decompres* or decod* or cod* or encoc Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **On ordering color maps for lossless predictive coding**
Memon, N.D.; Venkateswaran, A.;
Image Processing, IEEE Transactions on , Volume: 5 , Issue: 11 , Nov. 1996
Pages:1522 - 1527

[Abstract] [PDF Full-Text (620 KB)] **IEEE JNL**

---

2 **Adaptive lossy LZW algorithm for palettised image compression**
Chiang, S.W.; Po, L.M.;
Electronics Letters , Volume: 33 , Issue: 10 , 8 May 1997
Pages:852 - 854

[Abstract] [PDF Full-Text (496 KB)] **IEE JNL**

---

3 **Color image scalable coding with matching pursuit**
Figueras i Ventura, R.M.; Vandergheynst, P.; Frossard, P.; Cavallaro, A.;
Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). I International Conference on , Volume: 3 , 17-21 May 2004
Pages:iii - 53-6 vol.3

[Abstract] [PDF Full-Text (492 KB)] **IEEE CNF**

---

4 **JPEG2000 extensions for bit plane coding of floating point data**
Usevitch, B.E.;
Data Compression Conference, 2003. Proceedings. DCC 2003 , 25-27 March 2
Pages:451

[Abstract] [PDF Full-Text (204 KB)] **IEEE CNF**

IEEE Xplore® RELEASE 1.8

Welcome
United States Patent and Trademark Office

Try our New Full-text Search Prototype GO

Help

1) Enter a single keyword, phrase, or Boolean expression.
Example: acoustic imaging (means the phrase acoustic imaging plus any stem variations)
2) Limit your search by using search operators and field codes, if desired.
Example: optical <and> (fiber <or> fibre) <in> ti
3) Limit the results by selecting Search Options.
4) Click Search. See Search Examples

(compres* or decompres* or decod* or cod* or encod*) <paragraph> (color or colour) <paragraph> (lzw or

Start Search Clear

Note: This function returns plural and suffixed forms of the keyword(s).

Search operators: <and> <or> <not> <in> More

Field codes: au (author), ti (title), ab (abstract), jn (publication name), de (index term) More

Search Options:
Select publication types:

- [x] IEEE Journals
- [x] IEE Journals
- [x] IEEE Conference proceedings
- [x] IEE Conference proceedings
- [x] IEEE Standards

Select years to search:

From year: All to Present

Organize search results by:

Sort by: Relevance
In: Descending order

List 15 Results per page

Terms used
ompress or **decompress** or **encod** or **decod** **paragraph** **color** or **colour** **paragraph** **dictionary**

Found **1,278** of **147,793**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Optimizing encoding: Optimization of html automatically generated by wysiwyg programs

Jacqueline Spiesser, Les Kitchen

May 2004 **Proceedings of the 13th international conference on World Wide Web**

Full text available: pdf(129.59 KB) Additional Information: full citation, abstract, references, index terms

Automatically generated HTML, as produced by WYSIWYG programs, typically contains much repetitive and unnecessary markup. Thispaper identifies aspects of such HTML that may be altered whileleaving a semantically equivalent document, and proposes techniques to achieve optimizing modifications. These techniques include attribute re-arrangement via dynamic programming, the use of style classes, and dead-coderemoval. These techniques produce documents as small as 33% of original size. The size decre ...

Keywords: dynamic programming, haskell, html optimization, wysiwyg

2 HDR and perception: Perception-motivated high dynamic range video encoding

Rafal Mantiuk, Grzegorz Krawczyk, Karol Myszkowski, Hans-Peter Seidel

August 2004 **ACM Transactions on Graphics (TOG)**, Volume 23 Issue 3

Full text available: pdf(3.23 MB) Additional Information: full citation, abstract, references, index terms

Due to rapid technological progress in high dynamic range (HDR) video capture and display, the efficient storage and transmission of such data is crucial for the completeness of any HDR imaging pipeline. We propose a new approach for inter-frame encoding of HDR video, which is embedded in the well-established MPEG-4 video compression standard. The key component of our technique is luminance quantization that is optimized for the contrast threshold perception in the human visual system. The quant ...

Keywords: DCT encoding, HDR video, MPEG-4, adaptation, high dynamic range, luminance quantization, tone mapping, video compression, video processing, visual perception

3 Regular papers: On UNL as the future "html of the linguistic content" & the reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter

Gilles Sérasset, Christian Boitet

July 2000 **Proceedings of the 17th conference on Computational linguistics - Volume 2**

Full text available: pdf(571.35 KB) Additional Information: full citation, abstract, references

After 3 years of specifying the UNL (Universal Networking Language) language and prototyping deconverters from more than 12 languages and enconverters for about 4, the UNL project has opened to the community by publishing the specifications (v2.0) of the UNL language, intended to encode the meaning of NL utterances as semantic hypergraphs and to be used as a "pivot" representation in multilingual information and communication systems.A UNL document is an html document with special tags to delimi ...

Keywords: UNL, UNL-French localization, deconversion, generation, interlingua, pivot, transfer

4 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

5 Interactive semantic analysis of English paragraphs

Yorick Wilks

September 1969 **Proceedings of the 1969 conference on Computational linguistics**

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references

This paper describes the use of an on-line system to do word-sense ambiguity resolution and content analysis of English text paragraphs, using a system of semantic analysis programmed in Q32 LISP 1.5. The system of semantic analysis comprised dictionary codings for the text words, coded forms of permitted message, and rules producing message forms in combination on the basis of a criterion of semantic closeness. All these can be expressed within a single system of rules of phrase-structure form. ...

Keywords: interpretation, language analysis, semantics, template

6 Light field rendering

Marc Levoy, Pat Hanrahan

August 1996 **Proceedings of the 23rd annual conference on Computer graphics and interactive techniques**

Full text available: pdf(376.59 KB) Additional Information: full citation, references, citings, index terms

Keywords: epipolar analysis, holographic stereogram, image-based rendering, light field, vector quantization

7 Transformations and Experiences: VXT: a visual approach to XML transformations

Emmanuel Pietriga, Jean-Yves Vion-Dury, Vincent Quint

November 2001

Proceedings of the 2001 ACM Symposium on Document engineering

Full text available: pdf(165.99 KB) Additional Information: full citation, abstract, references, citings, index terms

The domain of XML transformations is becoming more and more important as a result of the increasing number of applications adopting XML as their format for data exchange or representation. Most of the existing solutions for expressing XML transformations are textual languages, such as XSLT or DOM combined with a general-purpose programming language. Several tools build on top of these languages, providing a graphical environment. Transformations are however still specified in a textual way using ...

Keywords: XML transformations, XSLT, circus, visual programming languages, zoomable user interfaces

8 Session 5: Perspectives on software evolution 2: Software evolution: let's sharpen the terminology before sharpening (out-of-scope) tools

Roland T. Mittermeir

September 2001 **Proceedings of the 4th International Workshop on Principles of Software Evolution**

Full text available: pdf(794.11 KB) Additional Information: full citation, abstract, references, index terms, review

Research on software evolution focuses on one hand on empirical investigations studying changes in long-living software systems, on the other hand on methods and tools, how such evolutionary behaviour of software can be controlled or supported.This paper departs from the observation that the empirical work and the tool- or methods-builder's work are quite often only obliquely related. Too often, the two camps depart from a token-semantics of the word *evolution.* However, by ignoring the pr ...

Keywords: software evolution, terminology

9 Survey of code-size reduction methods

Árpád Beszédes, Rudolf Ferenc, Tibor Gyimóthy, André Dolenc, Konsta Karsisto

September 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 3

Full text available: pdf(443.89 KB) Additional Information: full citation, abstract, references, index terms

Program code compression is an emerging research activity that is having an impact in several production areas such as networking and embedded systems. This is because the reduced-sized code can have a positive impact on network traffic and embedded system costs such as memory requirements and power consumption. Although code-size reduction is a relatively new research area, numerous publications already exist on it. The methods published usually have different motivations and a variety of appli ...

Keywords: code compaction, code compression, method assessment, method evaluation

10 Two bit/pixel full color encoding

Graham Campbell, Thomas A. DeFanti, Jeff Frederiksen, Stephen A. Joyce, Lawrence A. Leske

August 1986 **ACM SIGGRAPH Computer Graphics , Proceedings of the 13th annual conference on Computer graphics and interactive techniques**, Volume 20 Issue 4

Full text available: pdf(6.94 MB) Additional Information: full citation, abstract, references, citings, index terms

Realism in computer graphics typically requires using 24 or more bits/pixel to generate an

image. This paper describes a method developed by the authors called "Color Cell Compression" or "CCC" that preserves at least a limited animation and local update capability yet yields extraordinary-looking color images in approximately two bits/pixel independent of image complexity. Three intermediate methods of compressing images to six, four and three bits/pixel respectively are also described. The CCC ...

11 Compression of concordances in full-text retrieval systems

Y. Choueka, A. S. Fraenkel, S. T. Klein

May 1988 **Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval**

Full text available: pdf(1.36 MB) Additional Information: full citation, abstract, references, citings, index terms

The concordance of a full-text information retrieval system contains for every different word W of the data base, a list L(W) of "coordinates", each of which describes the exact location of an occurrence of W in the text. The concordance should be compressed, not only for the savings in storage space, but also in order to reduce the number of I/O operations, since the file is usually kept in secondary memory. Several methods are ...

12 Pen computing: a technology overview and a vision

André Meyer

July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

Full text available: pdf(5.14 MB) Additional Information: full citation, abstract, citings, index terms

This work gives an overview of a new technology that is attracting growing interest in public as well as in the computer industry itself. The visible difference from other technologies is in the use of a pen or pencil as the primary means of interaction between a user and a machine, picking up the familiar pen and paper interface metaphor. From this follows a set of consequences that will be analyzed and put into context with other emerging technologies and visions.Starting with a short historic ...

13 Automatic Subject Recognition in Scientific Papers: An Empirical Study

John O'Connor

October 1965 **Journal of the ACM (JACM)**, Volume 12 Issue 4

Full text available: pdf(1.65 MB) Additional Information: full citation, references, citings, index terms

14 The implementation and performance of compressed databases

Till Westmann, Donald Kossmann, Sven Helmer, Guido Moerkotte

September 2000 **ACM SIGMOD Record**, Volume 29 Issue 3

Full text available: pdf(129.75 KB) Additional Information: full citation, abstract, citings, index terms

In this paper, we show how compression can be integrated into a relational database system. Specifically, we describe how the storage manager, the query execution engine, and the query optimizer of a database system can be extended to deal with compressed data. Our main result is that compression can significantly improve the response time of queries if very *light-weight* compression techniques are used. We will present such light-weight compression techniques and give the results of runni ...

15 Storing text retrieval systems on CD-ROM: compression and encryption considerations

Shmuel T. Klein, Abraham Bookstein, Scott Deerwester

July 1989 **ACM Transactions on Information Systems (TOIS)**, Volume 7 Issue 3

Full text available: pdf(1.53 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The emergence of the CD-ROM as a storage medium for full-text databases raises the question of the maximum size database that can be contained by this medium. As an example, the problem of storing the Trésor de la Langue Fran&ccidel;aise on a CD-ROM is examined in this paper. The text alone of this database is 700 megabytes long, more than a CD-ROM can hold. In addition, the dictionary and concordance needed to access these data must be stored. A further constraint is that some of th ...

16 AI and computational logic and image analysis (AI): Symbol representation in map image compression

Akimov Alexander, Pasi Fränti

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available: pdf(536.30 KB) Additional Information: full citation, abstract, references, index terms

We propose map image compression system, in which we separate text and symbol information from the rest of the data. The text and other symbols are stored as one bitmap for each symbol into a dictionary. The technical challenge of the work is to convert the symbol data directly to output format similar to that of the JBIG2 standard. In this way, the text elements and special symbols are compressed more efficiently but we still have the maps in compatible raster image format.

Keywords: compression, map images, navigation, symbol representation

17 Technical Papers: A library of generic concepts for composing knowledge bases

Ken Barker, Bruce Porter, Peter Clark

October 2001 **Proceedings of the international conference on Knowledge capture**

Full text available: pdf(190.88 KB) Additional Information: full citation, abstract, references, citings, index terms

Building a knowledge base for a given domain traditionally involves a subject matter expert and a knowledge engineer. One of the goals of our research is to eliminate the knowledge engineer. There are at least two ways to achieve this goal: train domain experts to write axioms (*i.e.*, turn them into knowledge engineers) or create tools that allow users to build knowledge bases without having to write axioms. Our strategy is to create tools that allow users to build knowledge bases through ...

Keywords: knowledge engineering, knowledge reuse, ontologies

18 Session VI: Salience: the key to the selection problem in natural language generation

E. Jeffrey Conklin, David D. McDonald

June 1982 **Proceedings of the 20th conference on Association for Computational Linguistics**

Full text available: pdf(591.21 KB) Publisher Site Additional Information: full citation, abstract, references, citings

We argue that in domains where a strong notion of salience can be defined, it can be used to provide: (1) an elegant solution to the selection problem, i.e. the problem of how to decide whether a given fact should or should not be mentioned in the text; and (2) a simple and direct control framework for the entire deep generation process, coordinating proposing, planning, and realization. (Deep generation involves reasoning about conceptual and rhetorical facts, as opposed to the narrowly linguis ...

19 Multimedia document presentation, information extraction, and document formation in MINOS: a model and a system

S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, A. Pathria

December 1986 **ACM Transactions on Information Systems (TOIS)**, Volume 4 Issue 4

Full text available: pdf(3.16 MB) Additional Information: full citation, abstract, references, citings, index terms, review

MINOS is an object-oriented multimedia information system that provides integrated facilities for creating and managing complex multimedia objects. In this paper the model for multimedia documents supported by MINOS and its implementation is described. Described in particular are functions provided in MINOS that exploit the capabilities of a modern workstation equipped with image and voice input-output devices to accomplish an active multimedia document presentation and browsing within docu ...

20 Document formatting: Creating reusable well-structured PDF as a sequence of component object graphic (COG) elements

Steven R. Bagley, David F. Brailsford, Matthew R. B. Hardy

November 2003 **Proceedings of the 2003 ACM symposium on Document engineering**

Full text available: pdf(458.01 KB) Additional Information: full citation, abstract, references, index terms

Portable Document Format (PDF) is a page-oriented, graphically rich format based on PostScript semantics and it is also the format interpreted by the Adobe Acrobat viewers. Although each of the pages in a PDF document is an independent graphic object this property does not necessarily extend to the components (headings, diagrams, paragraphs etc.) within a page. This, in turn, makes the manipulation and extraction of graphic objects on a PDF page into a very difficult and uncertain process.The wo ...

Keywords: PDF, form Xobjects, graphic objects, tagged PDF

Search Results [PDF FULL-TEXT 496 KB] PREV NEXT DOWNLOAD CITATION

Adaptive lossy LZW algorithm for palettised image compression

Chiang, S.W. Po, L.M.
Dept. of Electron. Eng., City Univ. of Hong Kong, Kowloon
This paper appears in: **Electronics Letters**

Publication Date: 8 May 1997
On page(s): 852 - 854
Volume: 33 , Issue: 10
ISSN: 0013-5194
Reference Cited: 7
CODEN: ELLEAK
Inspec Accession Number: 5593073

Abstract:
An adaptive lossy **LZW** algorithm is proposed for palettised image **compress** a generalised algorithm of the conventional lossless **LZW** algorithm. The new employs an adaptive thresholding mechanism with human visual characteristi constrain the distortion. With this distortion control, the compression efficienc increased by ~40% for natural colour images, while maintaining good subject on the reconstructed image. In addition, the encoded image file format is con the original GIF decoder

Index Terms:
data **compression** image **coding** adaptive lossy **LZW** algorithm adaptive threshol mechanism **compression** efficiency distortion control human visual characteristics format natural **colour** images palettised image **compression** subjective quality

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Search Results [PDF FULL-TEXT 496 KB] PREV NEXT DOWNLOAD CITATION

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ | Terms | Back to Top

$$|\Delta V_{SB}| < (\phi + V_{SB0}) \quad (5)$$

and is obtained by applying the ratio test of convergence of a series [4]:

$$\lim_{n\to\infty}\left|\frac{W_{n+1}}{W_n}\right| = L \quad (6)$$

The series converges if $L < 1$, where

$$W_n = \frac{(-1)^n K_c \prod_{j=1}^{n-1}(2j-1)(\Delta V_{SB})^n}{2^{n-1}n!(\phi + V_{SB0})^{\frac{2n-1}{2}}}$$

$$W_{n+1} = \frac{(-1)^{n+1} K_c \prod_{j=1}^{n}(2j-1)(\Delta V_{SB})^{n+1}}{2^n(n+1)!(\phi + V_{SB0})^{\frac{2n+1}{2}}}$$

are the n^{th} and the $(n+1)^{th}$ term in the series, and $K_c = \gamma K_v K_\psi/(1+\delta)$

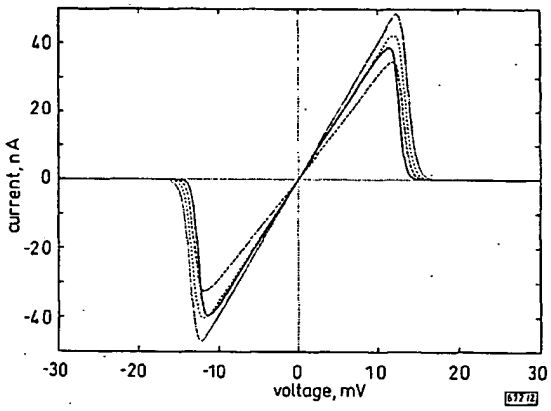

Fig. 2 *V-I characteristics of resistive fuse*

——— V1 = 0.1 V
- - - - V1 = 0.2 V
- - - - - V1 = 0.3 V
····· V1 = 0.4 V

The common-mode sensitivity of the circuit is overcome by using a MOS transistor *MN*5 having a higher resistance (lesser *W*/*L* ratio) in series with *MN*3 and *MN*4, and changing the gate voltage of *MN*5 so that its gate-source voltage remains constant over the range of operation. This is achieved by biasing *MP*3 so that its drain current is equal to the quiescent drain current of *MN*1 or *MN*2 and *MN*6 now acts as a voltage follower with the diode connected MOS transistor *MN*7 serving as a positive offset [5].

Results: The circuit shown in Fig. 1 was simulated using HSPICE with *W*/*L* ratio 5/5 for transistors *MN*1 and *MN*2 and 7/5 for *MP*1 and *MP*2. The other process parameters for a bulk-driven *n*-channel MOS transistor were $\phi = 0.725$V, $\gamma = 0.586$V$^{1/2}$, $V_{T0} = 0.828$V and $K' = 6.8 \times 10^{-5}$A/V^2. Fig. 2 shows the V-I characteristics of the resistive fuse shown in Fig. 1. The terminal *V*1 was fixed at a particular voltage and a ramp voltage was applied to the terminal *V*2. It is seen from the simulation results that V_{eff} is fairly constant for the voltage range 0.05–0.45V which satisfies $|\Delta V_{SB}| < (\phi + V_{SB0})$. The bulk-source junction is slightly forward biased and therefore the bulk current is negligable. The simulation results of a 2D array of a noisy 16 × 16 pixel synthetic image with voltage levels 0.1–0.4V depicting 0–64 levels of image intensity is also shown in Fig. 3.

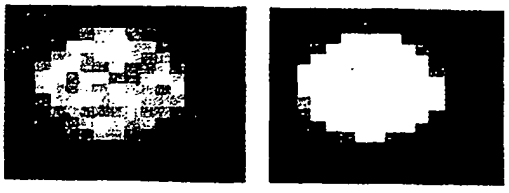

Fig. 3 *Test image and image after smoothing*

In summary, a resistive fuse operating on a 1.5V power supply is proposed. The range of operation can be increased by using the multi-threshold process to reduce threshold voltages of the transistors *MN*5 and *MN*7.

Acknowledgment: We wish to thank T. Manku, E.A. Abou-Allam, S.M. Franklin and L.A. MacEachern for their help in carrying out this research.

19 March 1997
Electronics Letters Online No: 19970569

J.K.V. Athreya and P.H. Gregson (*Computer Vision and Image Processing Laboratory, Department of Electrical Engineering, Technical University of Nova Scotia, Halifax, Nova Scotia B3J 2X4, Canada*)

References

1 PERONA, P., and MALIK, J.: 'Scale-space and edge detection using anisotropic diffusion', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990, 12, (7), pp. 629–639
2 YU, P.C., DECKER, S.J., LEE, H., SODINI, C.G., and WATT, J., Jr.: 'CMOS resistive fuses for image smoothing and segmentation', *IEEE J. Solid-State Circuits*, 1992, 27, (4), pp. 545–553
3 BLALOCK, B.J., ALLEN, P.E., and RINCON, G.A.: 'A 1 V CMOS opamp using bulk-driven MOSFETs'. Proc. IEEE Int. Solid-State Circuits Conf., 1995
4 KREYSZIG, E.: 'Advanced engineering mathematics' (Wiley Eastern Limited, 1994)
5 MEAD, C.: 'Analog VLSI and neural systems' (Addison-Wesley Publishing Company, 1989)

Adaptive lossy LZW algorithm for palettised image compression

S.W. Chiang and L.M. Po

Indexing terms: Data compression, Image processing

An adaptive lossy LZW algorithm is proposed for palettised image compression, that is a generalised algorithm of the conventional lossless LZW algorithm. The new algorithm employs an adaptive thresholding mechanism with human visual characteristics to constrain the distortion. With this distortion control, the compression efficiency is increased by ~40% for natural colour images, while maintaining good subjective quality on the reconstructed image. In addition, the encoded image file format is compatible with the original GIF decoder.

Introduction: The most popular palettised image compression format used in the World Wide Web (WWW) is the graphical interchange format (GIF). Its core compression algorithm is the well-known LZW algorithm (LZW) [1]. Since the LZW is a lossless compression algorithm, the compression efficiency of the GIF encoder is relatively low. Recently, lossy compression algorithms using DCT coding [2, 3], subband coding [4] and quadtree-based coding [5] are proposed for improving the compression efficiency of palettised image coding. However, the lossy DCT and subband coding techniques are designed for compression of continuous-tone images. The direct application of these techniques to palettised image compression may cause significant visual artefacts due to false assignment of colour, even for the colour ordering techniques used in [2 – 4]. In addition, a blocking artefact may also occur in the reconstructed image of the lossy quadtree-based method [5]. However, the most serious disadvantage of the these algorithms in practical applications is that they are incompatible with the GIF format. Thus, WWW browsers and many other applications embedded with GIF decoders cannot decode these compressed images. To tackle these problems, we propose a new lossy LZW algorithm using an adaptive threshold to enhance the compression performance of the GIF encoder. The proposed algorithm constrains the distortion according to the properties of the human visual system (HVS) in the encoding process, thus a visually lossless reconstructed image can be achieved.

Lossy LZW algorithm: The conventional LZW algorithm is a dictionary-based lossless compression method. Its major features are a fast encoding process, a high compression ratio and simple hardware implementation. Therefore, many commercial products are based on the LZW including ARC, PAK and the standard UNIX

compression tool 'COMPRESS'. To improve the performance of the lossless LZW algorithm for the palettised image compression, a new lossy LZW algorithm is proposed. The new algorithm generalises the original algorithm by introducing a distortion threshold T in the string matching process. When $T = 0$, the lossy LZW algorithm will be the same as the conventional lossless LZW algorithm. Since the pixels of the palettised image are colour indices of the palette (colour map), instead of physical intensity values, the matching process between two pixels should be applied to the corresponding colour vectors in the palette. In the proposed algorithm, we employ a weighted mean square error distortion measurement between the colour vectors. This distortion is defined in the YUV colour domain instead of the RGB domain, since the former is a more perceptually useful representation of colour for the HVS. Let i and j be the palettised image pixels with the corresponding colour vectors of $[R_i, G_i, B_i]$ and $[R_j, G_j, Bj]$, respectively. The colour vector of the index i in the YUV domain is given by $[Y_i, U_i, V_i]$ and its component values are determined by

$$\left.\begin{aligned} Y_i &= 0.30R_i + 0.59G_i + 0.11B_i \\ U_i &= 0.60R_i + 0.28G_i + 0.32B_i \\ V_i &= 0.21R_i + 0.52G_i + 0.31B_i \end{aligned}\right\} \quad (1)$$

The weighted mean square error distortion measure D_{ij} between two palettised image pixels of i and j is defined as

$$D_{ij} = \left[\frac{5}{8}(Y_i - Y_j)^2 + \frac{3}{16}(U_i - U_j)^2 + \frac{3}{16}(V_i - V_j)^2\right]^{\frac{1}{2}} \quad (2)$$

Based on this distortion measure, the lossy LZW for the GIF compression algorithm can be summarised as follows:

(i) All the colour vectors $[R_k, G_k, B_k]$ for $k = 0, 1, 2, ..., 255$ of the palette are transformed into the YUV domain colour vectors of $[Y_k, U_k, V_k]$ by eqn. 1.

(ii) The dictionary is initialised to contain the 256 colour indices (from 0–255). The first prefixed string ω is initialised to be the first input pixel of the image.

(iii) The next pixel i is obtained from the input stream. A full search is carried out in the dictionary for the string ωi while using the distortion D_{ij} as defined in eqn. 2 and a minimum D_{ij} is found for it.

(iv) If the minimum $D_{ij} \leq T$ for a certain adaptive threshold T, the two strings are considered as matched. The string ωi will be used as a prefixed string for the next iteration, i.e. $\omega i \rightarrow \omega$. Then go to (iii).

(v) If the minimum $D_{ij} > T$, the two strings are not matched. The code ω will be directed to the output stream and ωi will be pushed into the dictionary as a new entry. Then i will be used as the prefixed string, i.e. $i \rightarrow \omega$. Then go to (iii).

This process continues until the input is exhausted. For special cases, such as handling dictionary full and special input patterns, they are treated as those in the lossless LZW algorithm. This algorithm can easily be incorporated into the GIF format, thus the encoded image format is the same as the original GIF format. Therefore, the compatibility problem can be solved since the encoded images can be decoded by the conventional GIF decoder.

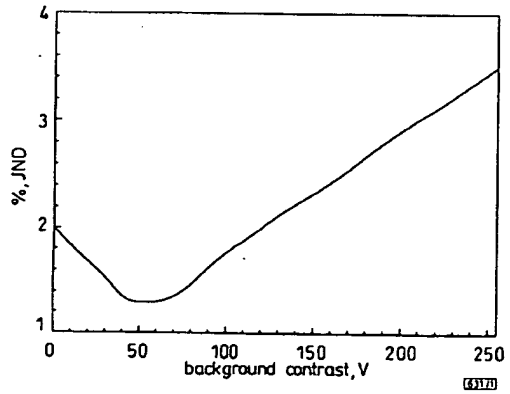

Fig. 1 *Weber's law of contrast*

Data are stored in adaptive threshold table

Adaptive threshold: To further enhance the compression efficiency of the lossy LZW, an adaptive threshold is used in the proposed algorithm which is designed according to the HVS properties. We propose use of the just noticeable difference (JND) stated in Weber's law [6]. Basically, the JND is the minimum amount of contrast difference between two luminance signals that the human eye can distinguish. Based on Weber's Law, the JND can be summarised as a function of the local luminance $\bar{Y}$ of a 3×3 subblock as shown in Fig. 1. This characteristic is highly nonlinear, thus a 256 look-up table is used to represent the relationship between the JND and the local luminance $\bar{Y}$ (i.e. $JND(\bar{Y})$ with $\bar{Y} = 0, 1, 2, ..., 255$). In addition, we also considered the spatial masking effects of the HVS. According to [7], two luminance signals will destructively mask each other when they coincide in an image. This effect is especially obvious when the two signals have similar frequencies.

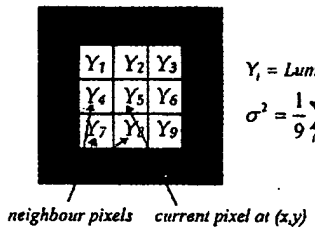

Fig. 2 *Spatial masking of HVS*

This is considered by monitoring local variance of image during encoding, leading to scaling factor

Therefore, high frequency noise generated during the encoding process will be masked when it is located in the high spatial frequency area of an image. The masking effect can be modelled as a function of a 3×3 subblock local variance σ^2 as shown in Fig. 2:

$$K = m\sigma^2 + b \quad (3)$$

where m and b are two constants. Combining the JND with the spatial masking effect, we can model the adaptive threshold as a function of the $JND(\bar{Y})$ and K:

$$T(JND(\bar{Y}), K) = JND(\bar{Y}) \cdot K = JND(\bar{Y}) \cdot (m\sigma^2 + b) \quad (4)$$

Fig. 3 *Y-component of reconstructed colour image Lena*

PSNR = 33.585, compression ratio = 1.7724

A GIF encoder using the proposed lossy LZW algorithm with an adaptive threshold is implemented. Simulation results for three 8 bit colour palettised images and two greyscale images are shown in Table 1. The values used for m and b in eqn. 3 are 0.027 and 0.9, respectively. The 8 bit colour palettised images are colour quantised from the standard 24bit images using a median-cut algorithm. The results in Table 1 show that the proposed algorithm achieved a 25–50% improvement incompression efficiency as compared with a conventional GIF encoder; it maintained a relatively high PSNR (29–35dB). In addition, a larger high compression

Table 1: Compression performances of lo[illegible]LZW and lossless LZW

Image (512 × 512)	Lossless LZW	Lossy LZW	PSNR
	% original	% original	dB
Lena	76.16	56.42	33.585
Airplane	52.52	33.53	34.896
Baboon	96.17	72.82	29.309
Lena (grey)	100.00	46.46	41.018
Mammo (grey)	66.01	26.15	42.885

ratio and PSNR are obtained for greyscale images. The reconstructed images of the original GIF and the lossy GIF encoders are observed from a distance of 4 times the image width. The two images are essentially indistinguishable and the Y-component of the reconstructed colour image of Lena is shown in Fig. 3.

Conclusions: In this Letter, a new lossy LZW algorithm for colour quantised image compression is proposed. HVS characteristics are used in the adaptive threshold to constrain the loss in the encoding process, resulting in a visually lossless reconstructed image. The GIF encoder with the proposed lossy LZW is implemented and tested with 8 bit colour quantised and greyscale images. A 25–50% improvement in the compression ratio is obtained for colour images, and that of 50–100% is obtained for greyscale images.

17 March 1997
Electronics Letters Online No: 19970558

S.W. Chiang and L.M. Po (*CityU Image Processing Laboratory, Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong*)

References

1 WELCH, T.A.: 'A technique for high-performance data compression', *IEEE Computer*, 1984, pp. 8–19
2 ZACCARIN, A., and LIU, B.: 'A novel approach for coding colour quantized image', *IEEE Trans. Image Process.*, 1993, 2, (4) pp. 442–453
3 CHEN, Y., PETERSON, H., and BENDER, W.: 'Lossy compression of palettized images'. Proc. ICASSP, 1993, Vol. 5, pp. 325–328
4 WALDEMAR, P., and RAMSTAD, T.A.: 'Subband coding of colour images with limited palette size'. Proc. ICASSP, 1994, Vol. 5, pp. V353–V356
5 PO, L.M., TAN, W.T., and WONG, W.B.: 'Quadtree based colour quantisation image compression', *Electron. Lett.*, 1995, **31**, (23), pp. 1988–1990
6 BORING, G.G.: 'A history of experimental psychology' (Appleton-Century-Crofts, New York, 1950), chap. 14, pp. 275–296
7 FOLEY, J.M., and LEGGE, G.E.: 'Contrast detection and near-threshold discrimination in human vision', *Vis. Res.*, 1981, **21**, pp. 1041–1053

Blocky artefact reduction using an adaptive constrained least squares method

Jeonghun Yang, Hyuk Choi and Taejeong Kim

Indexing terms: Image processing, Discrete cosine transforms, Image coding, Least squares approximations

A blocky artefact reduction algorithm using the constrained least squares (CLS) approach is described. The authors use a new objective function which effectively constrains the relationship between not only the block boundary pixels but also the inner pixels. By gracefully reducing the visible discontinuities along the block boundaries, the proposed algorithm shows excellent noise reduction performance.

Introduction: Block discrete cosine transform (BDCT) based coding schemes have been successfully applied to the compression of still and moving images. However, the blockwise operation of the coding algorithms causes undesirable blocky artefacts, i.e. visible grey-level discontinuities along the block boundaries. To reduce such artefacts, constrained least squares (CLS) based postprocessing schemes were recently proposed [1, 2]. CLS is a technique which minimises an objective function that imposes two conflicting requirements on the image being processed. However, the conventional approach which minimises only the block boundary difference of adjacent blocks still leaves the blocky artefacts hidden in the high frequency components of each block [2, 3]. To cope with this problem, we introduce a new objective function which simultaneously constrains both the block boundary differences and the inner differences. We propose a blocky artefact reduction algorithm that searches for the local CLS solution with respect to the new objective function.

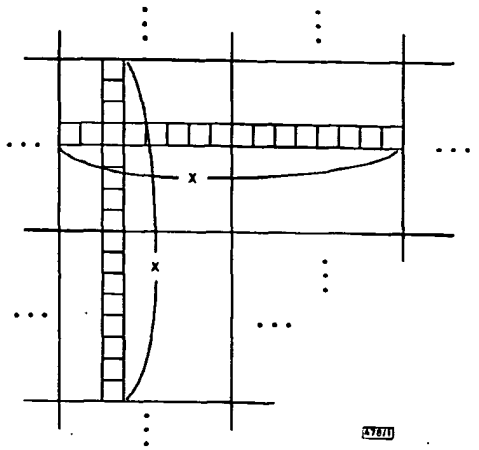

Fig. 1 *Formation of pixel vector*
$\mathbf{x} = (x(0)\ x(1)\ \ldots\ x(2N-1))^t$

New algorithm: To take advantage of the correlation between neighbouring blocks, we use one-dimensional pixel vectors along the neighbouring blocks as in Fig. 1. Let $\mathbf{x} = (x(0) \ldots x(2N-1))^t$ and $\hat{\mathbf{x}} = (\hat{x}(0) \ldots \hat{x}(2N-1))^t$ denote the pixel vectors both before and after postprocessing, respectively, when the image is encoded on an $N \times N$ block basis. In these vectors, $x(N-1)$ and $x(N)$ are called boundary pixels, and the other pixels except $x(0)$ and $x(2N-1)$ are considered as inner pixels. We also define the boundary difference as $x(N-1)-x(N)$ and the inner differences as $x(N-1+d)-x(N+d)$, $d = \pm 1$. From these pixel vectors, a local CLS solution $\hat{\mathbf{x}}$ is obtained by

$$\min_{\hat{\mathbf{x}}} \left(\|S\hat{\mathbf{x}}\|^2 + \mu \|\hat{\mathbf{x}} - \mathbf{x}\|^2 \right) \qquad (1)$$

S is a highpass operator that captures the highpass components of $\hat{\mathbf{x}}$ and is determined such that $\|S\hat{\mathbf{x}}\|$ penalises undesirable artefacts in the reconstruction. The term $\mu \|\hat{\mathbf{x}} - \mathbf{x}\|^2$ penalises the deviation

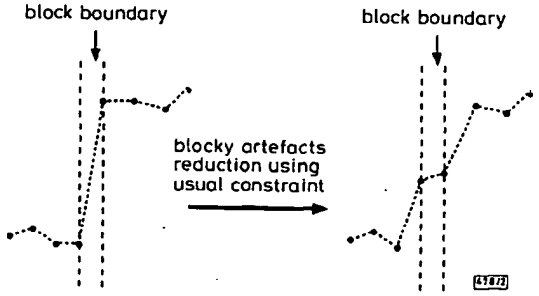

Fig. 2 *Transfer of blocky artefacts to inner pixels*
Adjusting boundary pixels enlarges inner differences

of the processed image from the received image, where μ defines the tradeoff between the smoothness of $\hat{\mathbf{x}}$ and the fidelity to the received data. However, without considering the inner pixels in $\|S\hat{\mathbf{x}}\|$, we can only recover the relationship between the block boundary pixels. Therefore, the CLS solution based solely on the boundary pixels is incapable of removing the blocky noise remaining within the respective blocks [2]. As an example, Fig. 2 shows

Context models for palette images

Ausbeck, P.J., Jr.
Dept. of Comput. Eng., California Univ., Santa Cruz, CA, USA;

This paper appears in: **Data Compression Conference, 1998. DCC '98. Pr**

Meeting Date: 03/30/1998 - 04/01/1998
Publication Date: 30 March-1 April 1998
Location: Snowbird, UT USA
On page(s): 309 - 318
Reference Cited: 8
Number of Pages: xvi+589
Inspec Accession Number: 6010841

Abstract:
A family of two dimensional context models appropriate for palette images is The models are designed for use with a binary arithmetic coder. A complete ir encoder/decoder using three models from the family is disclosed. The new co compared against five alternate coding methods: JBIG bit plane coding, CALI coding, CALIC plus palette ordering, and two dictionary methods, GIF and PN aggregate compression achieved by the new method on a corpus of fifteen pa images is 25% better than the best alternate method. The appropriateness of is validated by the similar aggregate compression achieved by the alternate n even though compression varies widely from image to image. Remarkably, th method achieves 20% better compression than a composite coder formed fro alternate method for each image

Index Terms:
arithmetic **codes** **codecs** digital arithmetic image **coding** image **colour** analysis constant techniques 2D context models CALIC predictive **coding** GIF JBIG bit pl PNG aggregate **compression** binary arithmetic **coder** **color** information compos **dictionary** methods image **encoder/decoder** lookup table palette images palette piecewise constant image model

Documents that cite this document
Select link to view other documents in the database that cite this one.

Context Models for Palette Images

Paul J. Ausbeck Jr.
Research Fellow
Department of Computer Engineering
University of California
Santa Cruz, CA 95064
paula@cse.ucsc.edu

Abstract

A family of two dimensional context models appropriate for palette images is described. The models are designed for use with a binary arithmetic coder. A complete image encoder/decoder using three models from the family is disclosed. The new coder is compared against five alternate coding methods: JBIG bit plane coding, CALIC predictive coding, CALIC plus palette ordering, and two dictionary methods, GIF and PNG. The aggregate compression achieved by the new method on a corpus of fifteen palette images is 25% better than the best alternate method. The appropriateness of the corpus is validated by the similar aggregate compression achieved by the alternate methods even though compression varies widely from image to image. Remarkably, the new method achieves 20% better compression than a composite coder formed from the best alternate method for each image.

1. Introduction

A palette image is composed of two components: color information contained in a lookup table or *palette*, and image information composed of a series of palette indices. In modern computer systems, palette images are ubiquitous. For example, the user interface elements of most windowing operating systems are composed of palette images. Black and white documents are a simple form of palette image. Almost every page on the Worldwide Web contains one or more palette images. Figure 1 is a grayscale version of the palette image serving as the banner of the Web site http://www.yahoo.com in October of 1997.

Figure 1
Typical Palette Image

In spite of their widespread use, an image model tailored for palette images has yet to be devised. Palette images generally contain too few colors to make effective use of the linear predictive models used in JPEG–LS and contain too many colors to avoid the sparse context problem that arises when using neighborhood color models such as those of JBIG. Table 1 shows the results of applying various coding methods to the grayscale image of Figure 1. The first method is to separately code the image bit planes with JBIG,

the second method is CALIC predictive coding[1], and the last method is dictionary coding via GIF. Perhaps surprisingly, the one dimensional LZW model used in GIF performs better than either of the alternate two dimensional models. For palette images that have not been converted to grayscale, the superior performance of dictionary coding methods continues to hold true. For typical palette images a text model turns out to perform better than the wrong image model!

Uncoded	Bit Planes	Predictive	GIF
27,182	9,266	8,348	6,923

Table 1
Motivational Coding Example

This paper proposes a new family of *piecewise–constant* image models that are optimized for describing palette images. The new models contain efficient mechanisms for describing pixel locations where color changes occur and a novel method for guessing unknown colors using known surrounding colors for probability estimation. The models are fully adaptive and have relatively few parameters.

Three specific members of the family that efficiently represent a wide range of palette images are also described. A new image coder based on these models is then compared to several alternate techniques on a corpus of typical palette images. Analysis and conclusions follow the experimental results.

2. A Palette Image Model

Whether synthetically produced or derived from continuous tone pictures, palette images are characterized by the following three properties.

- They tend to contain far fewer colors than pixels.
- Pixels of the same color tend to be contiguous.
- The color of a pixel is statistically related to its surrounding colors.

The first two palette image properties lead to a characterization of such images as *piecewise–constant*. For typical images, the constant color pieces or *domains* can be either rectilinearly or diagonally connected. If *boundaries* between domains are known, establishing the color of one pixel in a domain establishes the color of all pixels in the domain. One role of a piecewise–constant image model is to efficiently describe boundaries between arbitrarily shaped domains.

The statistical relationship of pixel colors propounded in property three is, for typical palette images, **not** a linear predictive relationship. Because of the lack of a simplifying model property, the sparse context problem makes it prohibitive to keep track of complete neighborhood color statistics. In an image with 256 colors, even a first order color model requires 256×255 model parameters. However, complete statistics are often unnecessary.

Given a first order context, typically only one or a few following colors predominate. This leads to the idea of using previously determined colors in the same context as *guesses*. To maintain one guess for each context of a 256 color first order model, requires only 256 model parameters.

Boundary delineation and color guessing can only determine whether or not an image pixel has the same color as some other pixel. Introduction of new colors or *innovations* into the image model requires some other mechanism. The exact form of this mechanism

is unrestricted and can be accomplished via a standard technique such as linear prediction.

A language for describing piecewise–constant images is shown in Table 2. It consists of a sequence of questions posed by a piecewise–constant image decoder. The questions are either binary decisions or can easily be decomposed into a sequence of binary decisions. As such, the language is designed for use with a binary arithmetic coder.

Q1	Is the current pixel's color identical to that of a specified rectilinearly connected neighbor?
Q2	Is the current pixel's color identical to that of a specified diagonally–connected neighbor?
Q3	Is the current pixel's color identical to a guessed value?
Q4	What is the current pixel's color?

Table 2
Piecewise–Constant Image Model Language Elements

The piecewise–constant modeling language is designed so that affirmative answers to **Q1–Q3** eliminate the need for a decoder to pose **Q4**. On images that match the model, **Q1–Q3** are answered predominantly in the affirmative and dominate both coding time and resulting codestring length. When **Q1–Q3** all fail the model reverts to the mechanism used to answer **Q4**.

Any mechanism can be used to answer **Q4** including linear predictive coding. In such a case the piecewise–constant model can be viewed as a preprocessor to a linear predictor. If adaptive context models are used for **Q1–Q3**, the worst case compression overhead introduced by the piecewise–constant model is largely proportional to the number of parameters used to estimate probabilities for **Q1–Q3**. The worst case performance overhead is proportional to the number of times that **Q1–Q3** are posed. Both of these topics are covered more extensively in the following discussion.

3. Context Models for the Piecewise–Constant Language

An excellent context model for **Q1** was proposed by Tate[2] in his work on coding edge maps resulting from image segmentation of satellite imagery. The technique is a variation of the neighborhood template model of Langdon[3]. Instead of using neighboring pixels on a black/white image, neighboring edge segments are used to determine a decision context. The construction is shown in Figure 2.

To completely specify an image partition, two edge locations are assigned to each pixel. For boundary discovery scans in normal raster order, either the north and west or the south and east edges are assigned to each pixel and the presence or absence of an edge segment is determined by a **Q1** decision. The following discussion defaults to north and west assignment.

For each location **L**, **Q1** is posed against the vertical dotted edge location on Figure 2 under the context determined by the eight solid edge segments on the figure. The western edge together with the same surrounding edges form a context to pose **Q1** against the

northern edge. The number of model parameters associated with this scheme is $256+512=768$, and the number of decisions is two per pixel.

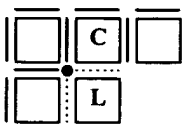

Figure 2
Rectilinear
Context

The number of model parameters and decisions of the Tate model can be reduced by taking advantage of boundary connectivity constraints. Since in a piecewise–constant model the boundaries separate contiguous domains, each end of a boundary segment must connect with another edge segment. Because of this constraint, many pixel locations require only one decision to determine domain connectivity.

For example, if the answer to **Q1** for the western edge of pixel **L** is yes, and no other edges impinge on the lattice intersection labeled with a dark circle associated with **L** on Figure 2, the answer to **Q1** for the northern edge of the location must always be yes and a coding decision need not be made. The average number of **Q1** decisions needed to establish rectilinear connectivity in an image varies between one and two decisions per pixel and for typical images approaches one. The connectivity constraints also decrease the number of model parameters associated with **Q1** to 512.

On bilevel images, connectivity constraints are even more severe. Since only two colors are available, a boundary lattice can never have an odd number of impinging edges, only zero, two or four. This constraint reduces the number of **Q1** decisions per pixel to one and the number of **Q1** model parameters to 256. Since the number of model parameters is halved on two color images, and since there are only two possible colors, better compression is achieved by including the color of one surrounding pixel in the context model. For example, if the color of the pixel labeled **C** on Figure 2 is added to the **Q1** context model, the number of model parameters is restored to 512. This formulation is equivalent to a nine pixel color model.

Q2 decisions are used to establish diagonal connectivity in a piecewise–constant model. Diagonal connectivity is only defined at lattice intersections where there is no rectilinear connectivity. Figure 3 shows the two causal orientations of diagonal connectivity at a lattice intersection, **L**, that has four impinging boundary segments. Each potential diagonal connection requires one **Q2** decision. The number of diagonal connections considered by the model drops quite rapidly as the edge density decreases and is typically less than 0.5 decisions per pixel.

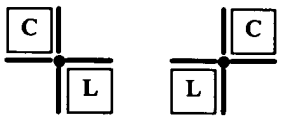

Figure 3
Diagonal Contexts

Domain color is usually more important than domain shape in conditioning **Q2** decisions. For this reason **Q2** decisions should only be made once the color to be propagated across an diagonal connection is known. Connection orientation is also an important conditioning criteria in many images. Using both orientation and color in a context model for **Q2** decisions requires two model parameters for every color used by an image. On Figure 3 the two orientations are represented by the left and right glyphs and the propagating color is labeled **C**.

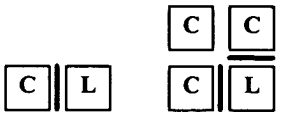

Figure 4
Guess Contexts

Color guessing, language element **Q3,** is designed to model the neighboring color relationships in an image while using a controlled number of model parameters. A guess is simply some color that has occurred previously in the coding process. The size of a guess model is proportional to D^S where D is the palette depth of the image and S is the number of neighboring colors used in the model. To maintain a reasonably sized model, the number of neighboring domain colors used to condition **Q3** must be carefully limited. For 256 color images it is usually only profitable to include one neighboring color in the conditioning context. The left glyph of Figure 4 shows a known pixel, **C**, used as a conditioning context for the unknown pixel, **L**, to its east. The right glyph of the figure shows three neighboring colors used as a conditioning context. The three color configuration is normally only useful for palette images of depth four (sixteen possible colors).

The exact size of a guess model is determined by the number of guesses for which statistics are maintained. One possibility is to maintain statistics for every possible color occurring in each context. The size of this straightforward guess model is D^{S+1}, no different from a complete neighborhood color model. With such a large model, many guesses are not very useful in determining color. Compression suffers because of the large number of mostly useless parameters to be learned. Coding speed suffers because a large number of mostly useless decisions are made.

One way to solve both of these problems is to limit the number of guesses maintained simultaneously by the model to some fixed number. When limiting guesses, a mechanism is needed to maintain only *good* guesses: guesses that are mostly correct. One way to achieve this is through *guess competition* in a *guess pool.*

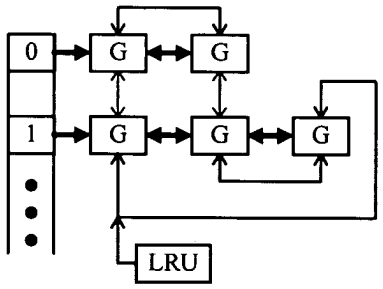

Figure 5
Guess Pool Structure

Any competitive mechanism can be used in the pool. One such mechanism is the least recently used (LRU) chain. In this application a context is moved to the front of the LRU any time its associated guess is correct. When a new guess is added to the pool and the pool has reached its maximum size, the guess at the end of the LRU chain is sacrificed. Figure 5 shows the guesses of a guess pool chained from lists determined by a context identifier. The average number of guesses per context is the size of the guess pool divided by the number of guess contexts. It should be emphasized that the guess pool is global. Competition occurs both between guesses in the same context and guesses in other contexts. The complete guess pool operation is as follows:

- When **Q4** decision is made, the result is added to the head of the guess pool LRU and to the tail of the appropriate decision context's guess chain.
- When a **Q3** decision is correct, its associated context is moved to the head of its guess chain and to the head of the guess pool LRU.

- When the guess pool is full and a guess must be sacrificed, it is removed from both its associated guess chain and the end of the LRU.
- The statistics of a **Q3** context are only updated when they are actually used to make a decision. No attempt is made to keep accurate conditional probabilities.
- The statistics of a sacrificed **Q3** context are decayed and kept as a prior for its new role.
- Guesses that are known to be impossible from prior **Q1** and **Q2** decisions are ignored.

4. An Example Piecewise–Constant Image Coder (PWC)

The piecewise–constant model components can be assembled in various ways to handle different types of images. The following discussion describes a coder that uses three variations on the piecewise–constant model to efficiently compress palette images of depth one, four, and eight. The number of parameters used by each model variation is summarized in Table 3.

Palette Depth	Model Parameters			
	Q1	**Q2**	**Q3**	**Q4**
1	512	0	0	1
4	512	0	256	64
8	512	512	1024	2048

Table 3
Model Parameters

The depth one model used by the example coder uses the single color augmented eight edge **Q1** context. The model does not use **Q2** and **Q3** since they provide no additional image information. **Q4** is only needed to determine the color of the first two domains in the image. Thereafter, color is propagated via color swap. Because **Q4** is only posed two times, the simplest **Q4** model, one parameter, is used.

At depth four, the eight edge model without a neighboring color is used for **Q1**. The three neighboring color model is used for **Q3**. Since this model has enough extent to provide most of the same information provided by **Q2**, **Q2** is not used. Although there are 4096 contexts, the size of the **Q3** guess pool is only 256, providing an average of 0.0625 guesses per context. Since on typical images the majority of possible neighboring color combinations never occur, the average number of guesses associated with each active context is much higher. A predictive model with four activity classes similar to that described by Don Speck[4] is used for **Q4**. The northern pixel is used for prediction and the magnitude of the maximum difference in the western, northwestern, and northern pixels is used to establish the activity class. The depth four coder performs a first raster scan and establishes rectilinear–connectivity via **Q1**. A second scan establishes color information via **Q3** and **Q4**.

At depth eight, the edge only model is again used for **Q1**. The orientation augmented single neighboring color model is used for **Q2**. The single neighboring color model with the neighboring color obtained from the western pixel is used for **Q3**. The size of the **Q3** context pool is 1024, providing four guesses per color on average. The predictive model used for **Q4** has eight activity classes. The prediction and activity class are established as with the depth four coder. The two pass operation of the coder is also the same.

It should be noted that for many images the **Q3** mechanism in the piecewise–constant model obviates the need for a sophisticated model for **Q4**. In fact in the experiments of the next section, a constant prediction in one activity class produced essentially the same

results as neighboring color prediction and multiple activity classes. The more sophisticated **Q4** mechanism only yields improvement if the image palette has significant linear predictive structure that is not caught by the piecewise–constant model.

5. Experiments

A standard corpus for evaluating palette image coding models is not currently available. Therefore significant effort was expended compiling a fairly representative group of test images. The resulting fifteen images are summarized in Table 4.

Image	Depth	Colors	Size	Source
benjerry	8	48	466x60	WWW – Ben and Jerry's ice cream ad
books	4	7	179x318	MS Access – bookshelf motif
ccitt01	1	2	1728x2339	CCITT – reference document #1
cmpndn	8	223	512x768	JPEG–LS cmpnd1 nearest color reduced
cmpndu	8	246	512x768	cmpndn plus error diffusion
flax	4	3	170x102	MS Access – textile simulation
gate	8	84	564x108	SF Chronicle home page – banner
music	4	8	111x111	MS Office clip art – music motif
netscape	8	32	612x100	Netscape home page – banner
pattern	1	2	135x137	MS Access – noisy tiled pattern
sea_dusk	8	46	484x325	MS Access – synthetic sky & city
stone	4	3	169x133	MS Access – stone texture
sunset	8	204	640x480	MS Scenes – sunset, ice and mountains
winaw	4	10	633x465	pcAnywhere – startup banner
yahoo	8	229	460x59	Yahoo home page – banner

Table 4
Palette Image Corpus

The number of colors used in the test images varies from two to 246. The test set contains completely synthetic images, error diffusion and nearest color quantized images, and compound images containing both synthetic and natural elements. An attempt was made to balance the number of bits of each source type. Some particular emphasis was placed on obtaining palette images from popular sites on the Worldwide Web. The **yahoo** image of line thirteen was shown previously in grayscale form in Figure 1.

Table 5 shows the results of the application of the piecewise–constant coder (PWC) described here and four alternate coding methods to the test images. All data are file byte counts and include header information. The header of the piecewise–constant file is 16 bytes long. The GIF, PNG, and PWC results contain either 6, 48, or 768 bytes of palette color information. The JBIG Planes and CALIC results do not.

Image	JBIG Planes	GIF	CALIC	CALICO	PNG	PWC
benjerry	3,697	4,401	4,193	4,193	4,571	3,018
books	13,343	11,177	14,033	9,740	10,831	8,624
ccitt01	12,884	38,862	18,146	18,146	28,910	12,892
cmpndn	69,098	62,682	56,951	52,085	56,397	40,790
cmpndu	81,705	76,759	71,109	63,430	69,438	54,550
flax	226	846	379	194	318	175
gate	25,589	23,313	20,555	18,494	20,124	16,318
music	1,563	1,987	1,648	1,219	1,647	744
netscape	16,489	17,442	14,302	13,746	15,879	11,326
pattern	1,190	1,782	1,755	1,755	1,928	1,203
sea_dusk	1,647	6,362	1,446	1,069	2,540	1,289
stone	4,460	4,753	8,440	4,917	3,906	3,992
sunset	121,020	100,186	113,710	82,109	81,794	52,608
winaw	27,830	18,559	21,686	16,886	18,732	11,439
yahoo	8,126	7,097	6,884	6,884	6,275	4,492
total	388,867	376,208	355,237	294,867	323,290	223,460

Table 5
Comparison of Coding Methods

The results in the column labeled JBIG Planes were obtained by applying JBIG[5] to the bit planes of each image. Empty and duplicate planes were eliminated from each image total. For each image only enough planes were kept to uniquely determine the number of colors used. To correct for duplicate file headers, eighty bytes were subtracted from each JBIG total for each retained plane other than the first.

The third column of results were obtained using the arithmetic CALIC[1] predictive coder. Of the JPEG–LS contributors, CALIC appears best suited to palette images. The publicly available implementation obtained from http://www.csd.uwo.ca/~wu/index.html was used. The results labeled CALICO were obtained by combining CALIC with palette ordering[6]. The GIF and PNG results were obtained using the Paint Shop Pro image utility[7].

Image	JBIG Planes	GIF	CALIC	CALICO	PNG	PWC
pc	123,788	376,482	205,000	186,157	225,898	98,997

Table 6
Coding Results for the JPEG–LS **pc** Image

The data of Table 6 are coding results for a palletized version of the JPEG–LS test image, **pc**. They are shown because **pc** is a widely known test image, contains only six colors, and can be converted to a depth four palette image without loss. This image is so large that it would skew the corpus, so it is shown separately. The proposed JPEG–LS standard with palette feature codes this image using 322,628 bytes. JBIG bit planes does well on this image but is still 20% worse than PWC.

Table 7 shows some key PWC coding statistics for the fifteen images of the test corpus. The column labeled "Domains" is a count of the number of rectilinearly connected domains in each image. The columns labeled with **Q?** are counts of the number of each type of decision made during coding. The columns labeled H? are the average entropies for each decision type.

As described previously, **Q2** decisions are only made for depth eight images. **Q3** decisions are only made for depths greater than one. Additionally, depth one images require only two **Q4** decisions. Due to the method used to obtain the measurements, the accuracy of the tabulated entropy values degrades substantially for those decision buckets containing fewer than several hundred decisions.

Image	Domains	**Q1**	H1	**Q2**	H2	**Q3**	H3	**Q4**	H4
benjerry	1704	32647	0.330	1022	0.845	7257	0.645	257	6.008
books	14527	84877	0.663	0	0.000	18418	0.593	453	2.808
ccitt01	1513	4041791	0.025	0	0.000	0	0.000	2	1.000
cmpndn	34445	462288	0.324	40380	0.592	155665	0.471	10603	6.873
cmpndu	61019	486937	0.297	88124	0.690	247177	0.480	15554	6.825
flax	16328	33906	0.018	0	0.000	16328	0.011	14	4.571
gate	15497	87841	0.625	19954	0.725	79404	0.462	2916	6.255
music	948	15244	0.283	0	0.000	1082	0.665	111	3.532
netscape	10385	83235	0.571	10862	0.769	36975	0.694	557	5.056
pattern	7142	28133	0.319	0	0.000	7137	0.010	7	4.571
sea_dusk	211	158628	0.020	4	1.000	1067	0.330	60	7.067
stone	6827	36070	0.793	0	0.000	7687	0.347	41	3.317
sunset	36925	390890	0.637	39594	0.664	122527	0.698	7158	7.540
winaw	9984	321561	0.255	0	0.000	11956	0.628	504	3.000
yahoo	3002	33519	0.444	3581	0.809	9425	0.609	815	7.539

Table 7
PWC Coding Statistics

6. Performance

Using a research grade implementation, the PWC encoded images of the previously described test set are decoded in 8 seconds on a 200 MHz Intel Pentium processor. This is approximately 225 Kbit/sec decode performance, sufficient to keep up with a 128 Kbit/sec ISDN connection. Encoding is about 25% slower due to increased memory use.

7. Conclusion

A new piecewise–constant image model and a language for describing such models has been proposed. The model is appropriate for application to palette image coding. An example piecewise–constant image coder demonstrated remarkable compression on a corpus of 15 palette images. The demonstrated compression is 20% better than a composite coder formed by selecting the best of JBIG bit plane coding, CALIC predictive coding with palette ordering, and PNG LZ77 deflation for each image of the corpus. The

appropriateness of the corpus was validated by the similar aggregate compression achieved by the competing methods even though their performance varied widely on individual images of the corpus.

The data show that the PWC coder is robust across a wide variety of images. The compression achieved is as good (2% worse on stone) or better than the best of the alternate methods on every image in the test set. Its aggregate compression is 31% better than PNG, and 24% better than the combination of CALIC with palette ordering. The other alternate methods are even less competitive.

8. Acknowledgments

The arithmetic coder used in the example coder is the carry free coder designed and written by Don Speck[4]. The Golomb tree coder[8] used for coding prediction errors was also done by Don. Nasir Memon graciously provided the palette ordering code used in the comparison experiments.

9. References

1 Xiaolin Wu, "An Algorithmic Study on Lossless Image Compression", Data Compression Conference Proceedings, March 31–April 3, 1996, IEEE Computer Society Press, Los Alamitos, California.

2 Stephen R. Tate, Lossless Compression of Region Edge Maps, CS–1992–9, Department of Computer Science, Duke University, Durham, NC, 1992.

3 Glen G. Langdon, Jr. and Jorma Rissanen, "Compression of Black–White Images with Arithmetic Coding", IEEE Transactions on Communications, Vol. COM–29(6), pp. 858–867 (June 1981).

4 Don Speck, "Local Activity Level Classification Model for Continuous–tone Coding", document N198 submitted to ISO/IEC JTC1/SC29/WG1 June 29, 1995.

5 Markus Kuhn, Version 0.9 of the JBIG–KIT, available via anonymous ftp at ftp.informatik.uni–erlangen.de/pub/doc/ISO/JBIG/jbigkit-0.8.tar.gz.

6 N. D. Memon and A. Venkateswaran, "On Ordering Color Maps for Lossless Predictive Coding", IEEE Transactions on Image Processing, 1996, Vol. 5, No. 11, pp. 1522–1527.

7 Jasc, Inc., http://www.jasc.com.

8 Don Speck, "Clarifying Some Details of ALCM", June 1996, document N351 submitted to ISO/IEC JTC1/SC29/WG1.